

Цель

Задание 1 состоит в численном решении антагонистической матричной игры. В рамках данного задания мы должны написать код, решающий матричную игру путем сведения ее к паре двойственных задач линейного программирования, проиллюстрировать работу данного кода путем визуализации спектров оптимальных стратегий. Цель задания 1 заключается в том, чтобы познакомиться с языком программирования Python, библиотекой SciPy и интерактивной средой разработки Jupyter.

Краткое описание задания

1. (50 баллов) Необходимо написать функцию `nash_equilibrium(a)`, которая принимает матрицу выигрыша и возвращает значение игры и оптимальные стратегии первого и второго игроков.
2. (50 баллов) Проиллюстрировать работу вашего кода путем решения нескольких игр и визуализации спектров оптимальных стратегий игроков в Jupyter. В частности, нужно привести игры, в которых:
 - i. спектр оптимальной стратегии состоит из одной точки (т.е. существует равновесие Нэша в чистых стратегиях),
 - ii. спектр оптимальной стратегии неполон (т.е. некоторые чистые стратегии не используются),
 - iii. спектр оптимальной стратегии полон.

Подход к решению

1. Написание функции `nash_equilibrium(a)`
 - На вход подается матрица выигрыша, реализующая антагонистическую матричную игру
 - Задача сводится к решению пары двойственных задач линейного программирования
 - Решение задач находим с помощью симплекс-метода, который позволяет получить на выходе оптимальные стратегии игроков и значение игры
2. Написание части, реализующей визуализацию спектров оптимальных стратегий игроков в Jupyter

- Необходимо использовать объекты и методы из библиотеки `matplotlib.pyplot` и подобрать необходимые матрицы, чтобы визуализировать решение с помощью гистограммы

Аналитическое пояснение

Необходимо решить антагонистическую матричную игру. **Игрой** с математической точки зрения называется тройка $\langle X, Y, F(x, y) \rangle$, где $X = \{X_1, \dots, X_n\}$ и $Y = \{Y_1, \dots, Y_n\}$ - множества стратегий первого и второго игроков соответственно, а $F(x, y)$ - это **функция выигрыша**. Матрица, в которой строки соответствуют стратегиям игрока X , а столбцы - стратегиям игрока Y , элементами $a(i, j)$ которой являются результаты игры при i -й стратегии игрока X и j -й стратегии игрока Y , называется **матрицей выигрышей**. Матричная игра называется **антагонистической**, если в ней участвуют два игрока и значения функций выигрыша в каждой ситуации равны по величине, но противоположны по знаку (весь проигрыш одного игрока является выигрышем другого). Нужно найти значение игры и оптимальные стратегии первого и второго игроков при заданной матрице.

Ход решения

Чистая стратегия - возможный ход игрока, выбранный с вероятностью 1. Для каждого игрока возникает вопрос увеличения выигрыша (уменьшения проигрыша). Это можно сделать применяя **смешанные стратегии** - вероятностное распределение на множестве X для первого игрока (Y - для второго).

1. Не ограничивая общности, приведем матрицу к удобному виду: если в ней есть отрицательные элементы, то прибавив модуль наименьшего элемента матрицы, мы получаем неотрицательную матрицу. Тогда мы можем сказать, что **цена игры** положительна: $v > 0$.
2. Рассмотрим смешанные стратегии $p = (p_1, \dots, p_m)$ и $q = (q_1, \dots, q_n)$, где компоненты этих векторов - вероятность применения чистых стратегий i и j первым и вторым игроками соответственно. $p_1 + p_2 + \dots + p_m = 1$;
 $q_1 + q_2 + \dots + q_n = 1$.
3. Найдем оптимальную стратегию первого игрока. Она должна обеспечить ему выигрыш $\geq v$ при любом поведении второго игрока. Значит, должна выполняться система неравенств:

$$\begin{aligned} a_{11} p_1 + a_{21} p_2 + \dots + a_{m1} p_m &\geq v \\ a_{12} p_1 + a_{22} p_2 + \dots + a_{m2} p_m &\geq v \\ &\dots \\ a_{1n} p_1 + a_{2n} p_2 + \dots + a_{mn} p_m &\geq v \end{aligned}$$

Разделим все неравенства на v :

$$\begin{aligned} a_{11} p_1/v + a_{21} p_2/v + \dots + a_{m1} p_m/v &\geq 1 \\ a_{12} p_1/v + a_{22} p_2/v + \dots + a_{m2} p_m/v &\geq 1 \end{aligned}$$

$$\dots$$

$$a_{1n} p_1/v + a_{2n} p_2/v + \dots + a_{mn} p_m/v \geq 1$$

Заменив p_i/v на y_i , $i = 1, 2, \dots, m$ получим задачу линейного программирования для первого игрока:

$$\min \leftarrow F(x, y^*) = y_1 + y_2 + \dots + y_m$$

$$y = (y_1, \dots, y_m)$$

$$a_{11} y_1 + a_{21} y_2 + \dots + a_{m1} y_m \geq 1$$

$$a_{12} y_1 + a_{22} y_2 + \dots + a_{m2} y_m \geq 1$$

$$\dots$$

$$a_{1n} y_1 + a_{2n} y_2 + \dots + a_{mn} y_m \geq 1$$

Аналогично получим задачу для второго игрока:

$$\max \leftarrow F(x^*, y) = x_1 + x_2 + \dots + x_m$$

$$x = (x_1, \dots, x_m)$$

$$a_{11} x_1 + a_{21} x_2 + \dots + a_{m1} x_m \leq 1$$

$$a_{12} x_1 + a_{22} x_2 + \dots + a_{m2} x_m \leq 1$$

$$\dots$$

$$a_{1n} x_1 + a_{2n} x_2 + \dots + a_{mn} x_m \leq 1$$

Для того, чтобы найти решение в смешанных стратегиях используем функцию linprog, которая с помощью **симплекс-метода** находит искомое решение.

Суть метода

Алгоритм симплекс-метода заключается в том, что из множества вершин, принадлежащих границе множества решений системы неравенств, выбирается такая вершина, в которой значение целевой функции достигает максимума (минимума). По определенному правилу находится начальный опорный план (нулевое решение) и проверяется на оптимальность. Если план оптимален, то задача решена. Если нет, то переходим к другой вершине. Значение целевой функции будет заведомо лучше, чем в предыдущей. Переход от одной вершины к другой осуществляется с помощью вычислений, которые удобно записывать в виде так называемых симплекс-таблицами. Так как вершин конечное число, то и к оптимальному решению мы приходим за конечное число шагов.

Шаги:

- Шаг 1. Составление симплекс-таблицы.** Перейдем к канонической форме задачи линейного программирования. Запишем ее в симплекс-таблицу. Количество строчек в таблице соответствует количеству равенств в системе ограничений, а столбцов - количеству свободных переменных. Базисные переменные заполняют первый столбец, свободные - верхнюю строку таблицы. Нижняя строка называется индексной, в ней записываются коэффициенты при переменных в целевой функции. В правом нижнем углу первоначально записывается 0, если в функции нет свободного члена; если есть, то он записывается с противоположным знаком. На этом месте (в правом нижнем углу) будет значение целевой функции, которое при переходе от одной таблицы к другой должно увеличиваться по модулю.

- **Шаг 2. Проверка на оптимальность.** Проверяем на положительность элементы строки целевой функции. Если найдется хотя бы один отрицательный коэффициент индексной строки, то план необходимо улучшить: из отрицательных коэффициентов индексной строки выбирается наибольший по абсолютной величине. Затем элементы столбца свободных членов симплексной таблицы делит на элементы того же знака ведущего столбца. Далее идет построение нового опорного плана. Пересчет симплекс-таблицы осуществляется методом Жордана—Гаусса. Если в нижней строке и в столбце свободных членов все элементы положительные, то *оптимальное решение найдено*.